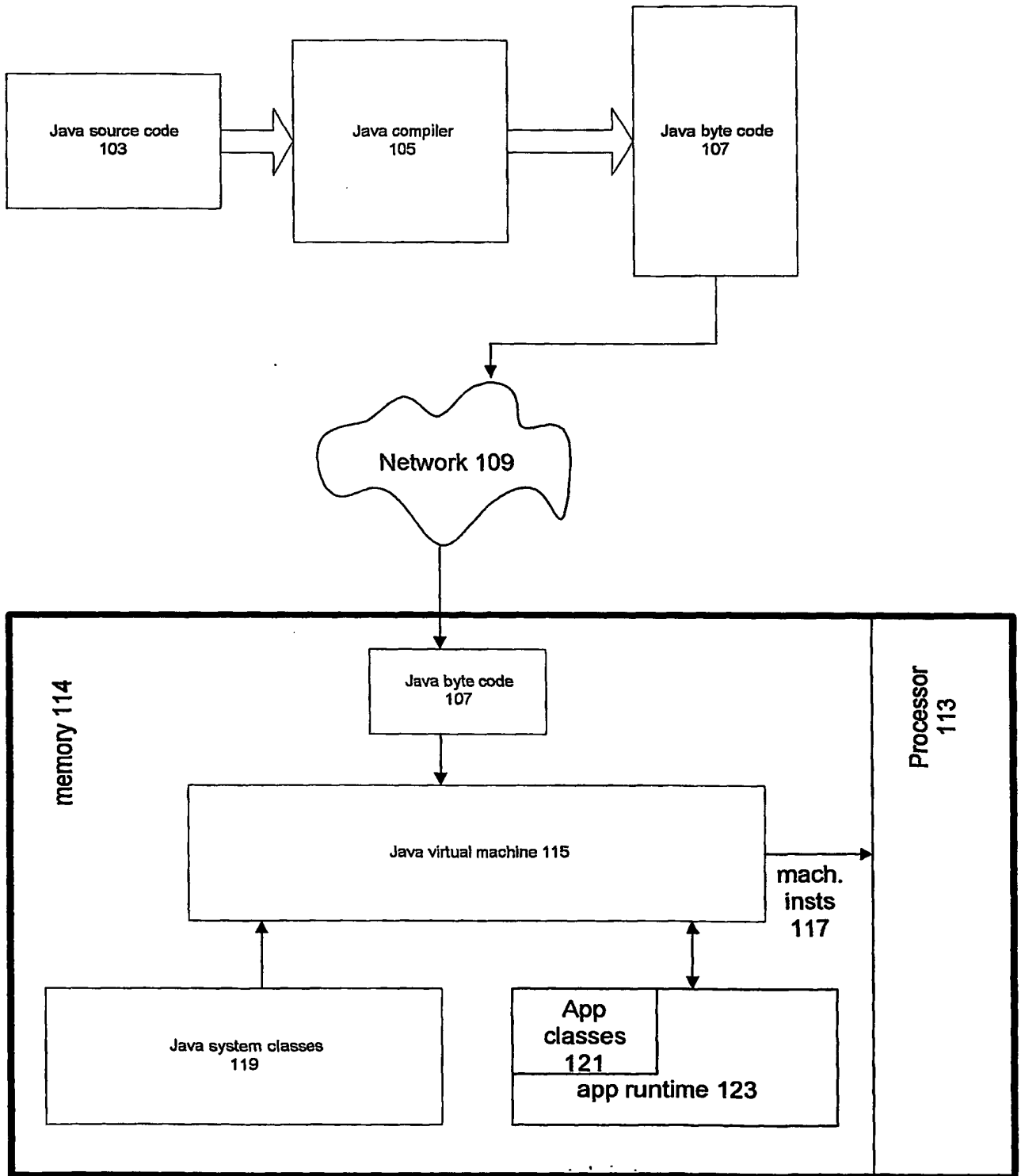


BEST AVAILABLE COPY



101

programmable device 111

Fig. 1

```

import java.util.Date;
public class Person {
    private String name;
    private Date dob; // date of birth
    public Person(String n, Date d) // constructor
    {
        name = n;
        dob = d;
    }
    public void changeName(String newName)
    {
        name = newName;
    }
}

```

201

Fig. 2

```

import java.util.Date;
import java.util.Vector; 303

public class Person {
    private Vector v = new Vector(); 304
    public Person(String n, Date d) // constructor
    {
        v.addElement(n);
        v.addElement(d); } 305
    }
    public void changeName(String newName)
    {
        String s = (String)v.elementAt(0); 307
        s = newName;
        v.setElementAt(s, 0); 309
    }
}

301

import java.util.Date;
import java.util.Vector;

public class P { 311
    private Vector v = new Vector();
    public P(String n, Date d) // constructor
    {
        v.addElement(n);
        v.addElement(d);
    }
    public void c(String n) 313
    {
        String s = (String)v.elementAt(0);
        s = n; 315
        v.setElementAt(s, 0);
    }
}

310

```

Fig. 3

```

public class P { 403
    private V v = new 405
    public P(S n, D d) // constructor
    {
        v.a(n); 407
        v.a(d); 411
    }
    public void c(S n) 413
    {
        S s = (S)v.b(0);
        s = n;
        v.c(s, 0); 415
    }
}

```

401

```

java.lang.String    S
java.util.Date      D
java.util.Vector     V

```

409

Fig. 4

```
Person p = new Person("John Doe", new Date());
p.changeName("John Hancock");
```

501

```
Class c = p.getClass(); 505
Method[] m = c.getMethods(); 507
String[] a = {"John Hancock"}; 509
m[1].invoke(p, a); // same as p.changeName("John
Hancock") 511
```

503

Fig. 5

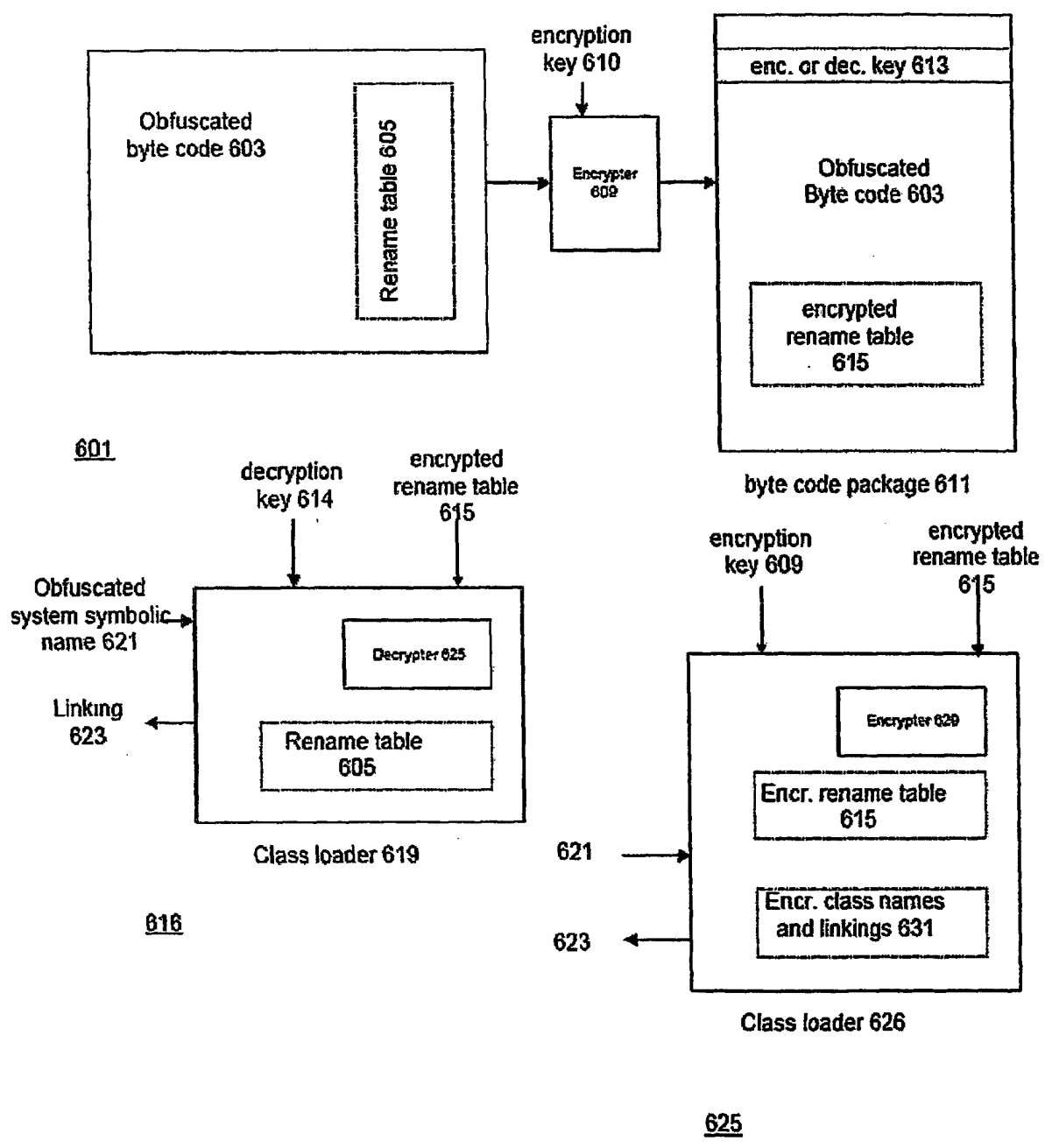


Fig. 6

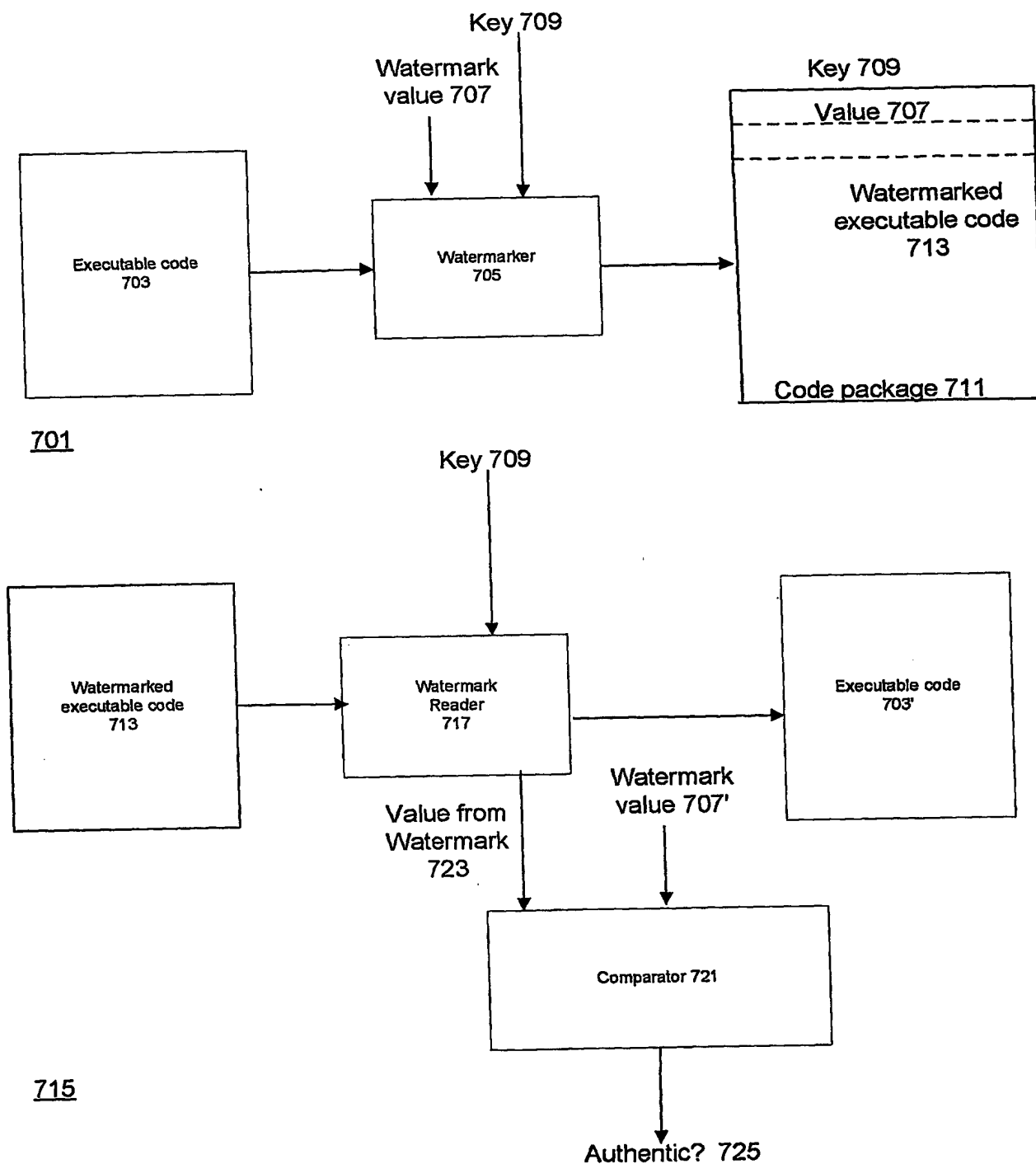


Fig. 7

```

private void add()
0:   iconst_0
1:   istore_2
2:   iconst_1
3:   istore_1
4:   goto      #14
7:   iload_2
8:   iload_1
9:   iadd
10:  istore_2
11:  iinc      %1    1
14:  iload_1
15:  bipush    100
17:  if_icmple #7
20:  getstatic  java.lang.System.out
      Ljava/io/PrintStream; (48)
803(a) 23:  nop
24:  new        <java.lang.StringBuffer> (32)
27:  dup
28:  ldc        "1 + 2 + ... + 100 = " (3)
30:  nop
803(b) 31:  invokespecial java.lang.StringBuffer.<init>
      (Ljava/lang/String;)V (38)
34:  iload_2
35:  invokevirtual java.lang.StringBuffer.append
      (I)Ljava/lang/StringBuffer; (40)
38:  ldc        "" (1)
803(c) 40:  nop
41:  invokevirtual java.lang.StringBuffer.append
      (Ljava/lang/String;)Ljava/lang/StringBuffer; (41)
44:  invokevirtual java.lang.StringBuffer.toString
      ()Ljava/lang/String; (52)
47:  invokevirtual java.io.PrintStream.println
      (Ljava/lang/String;)V (49)
50:  return

```

801

Fig. 8

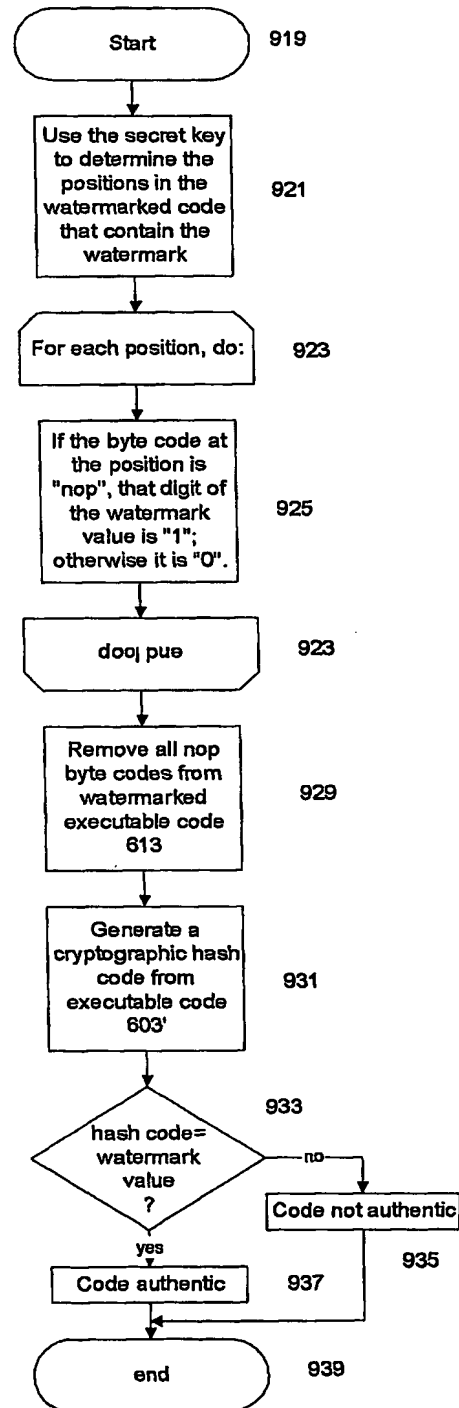
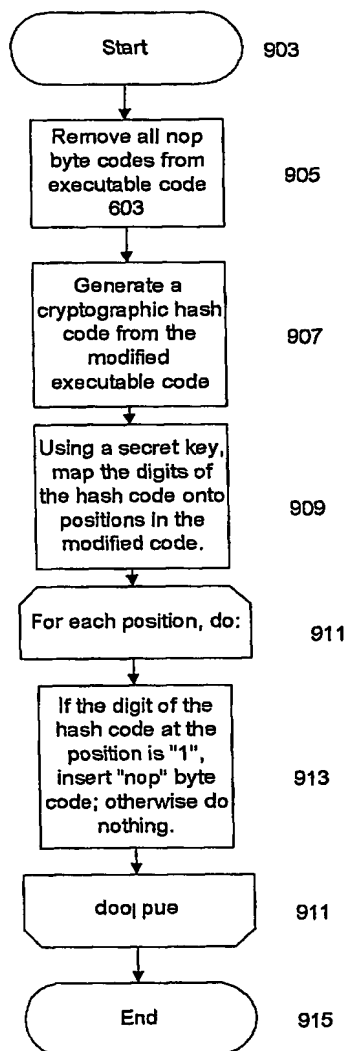


Fig. 9

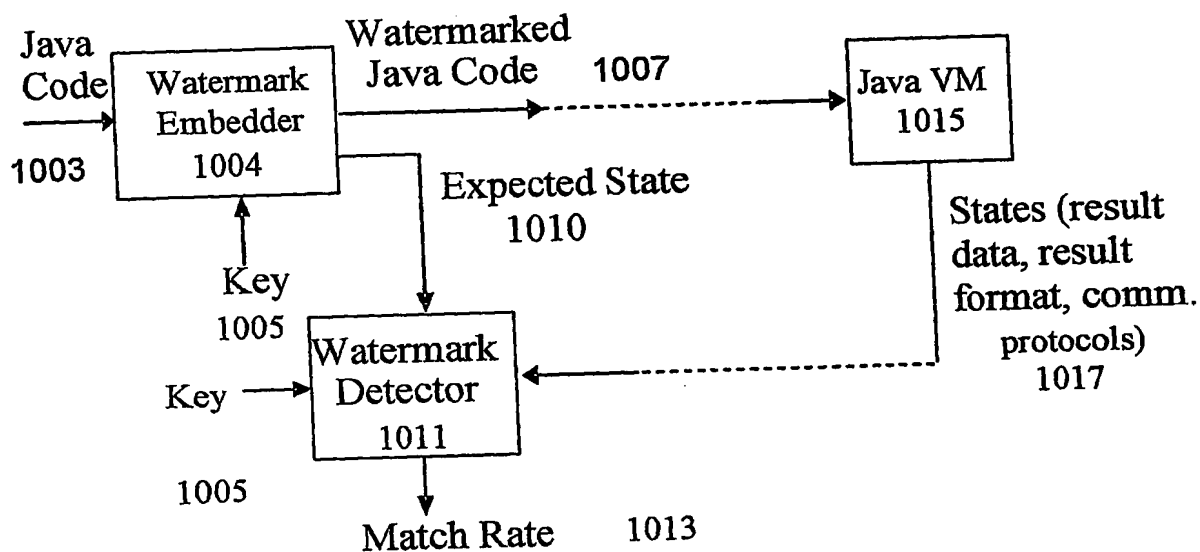


Fig. 10

```

12 putfield #250 <Field java.util.Vector nextSites>
15 aload_0
16 ldc #48 <String "Failed: ">
18 putfield #210 <Field java.lang.String failures>
21 aload_0
22 aload_1
23 putfield #291 <Field DWASuitCase suitcase>
26 aload_0
27 aload_1
28 iconst_4
29 invokevirtual #217 <Method java.lang.Object get(int)>
32 checkcast #119 <Class DWAPolicy>
35 putfield #255 <Field DWAPolicy policy>
38 aload_0
39 aload_1
40 iconst_5
41 invokevirtual #217 <Method java.lang.Object get(int)>
44 checkcast #124 <Class DWASiteInfo>
47 putfield #286 <Field DWASiteInfo siteInfo>

```

1101

Fig. 11



Fig. 12

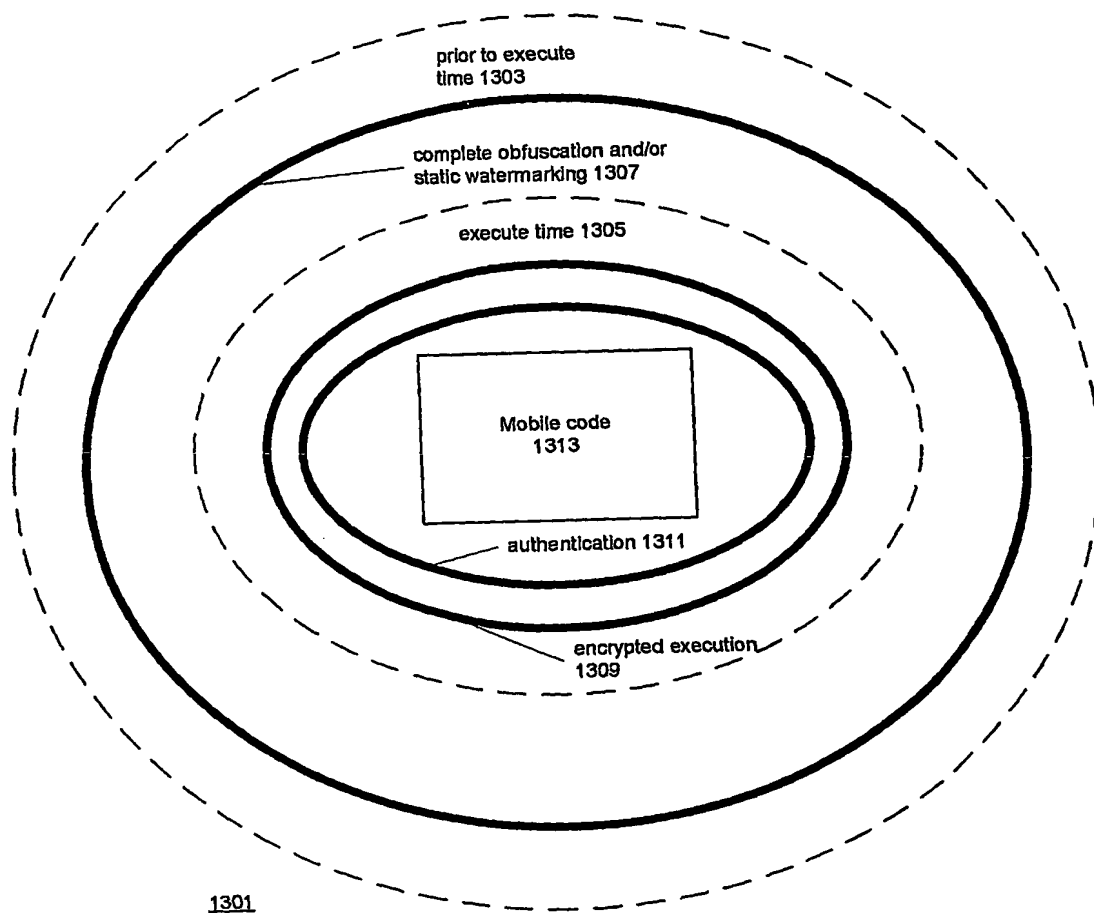


Fig. 13

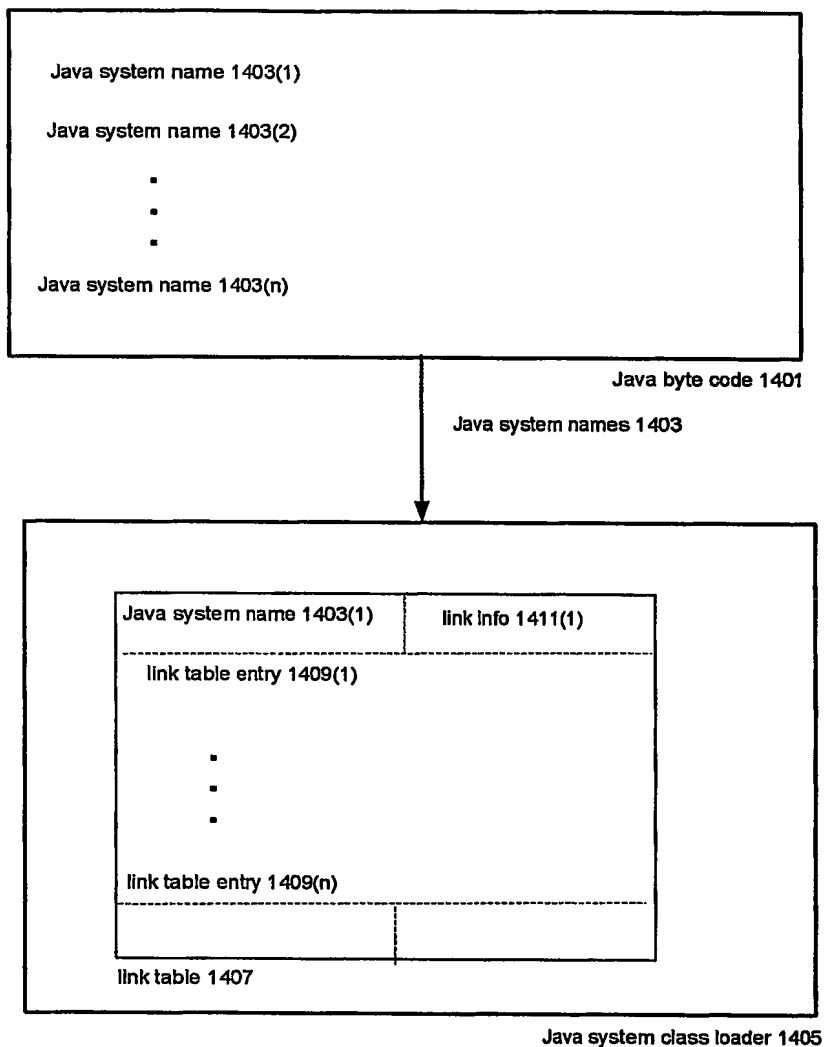
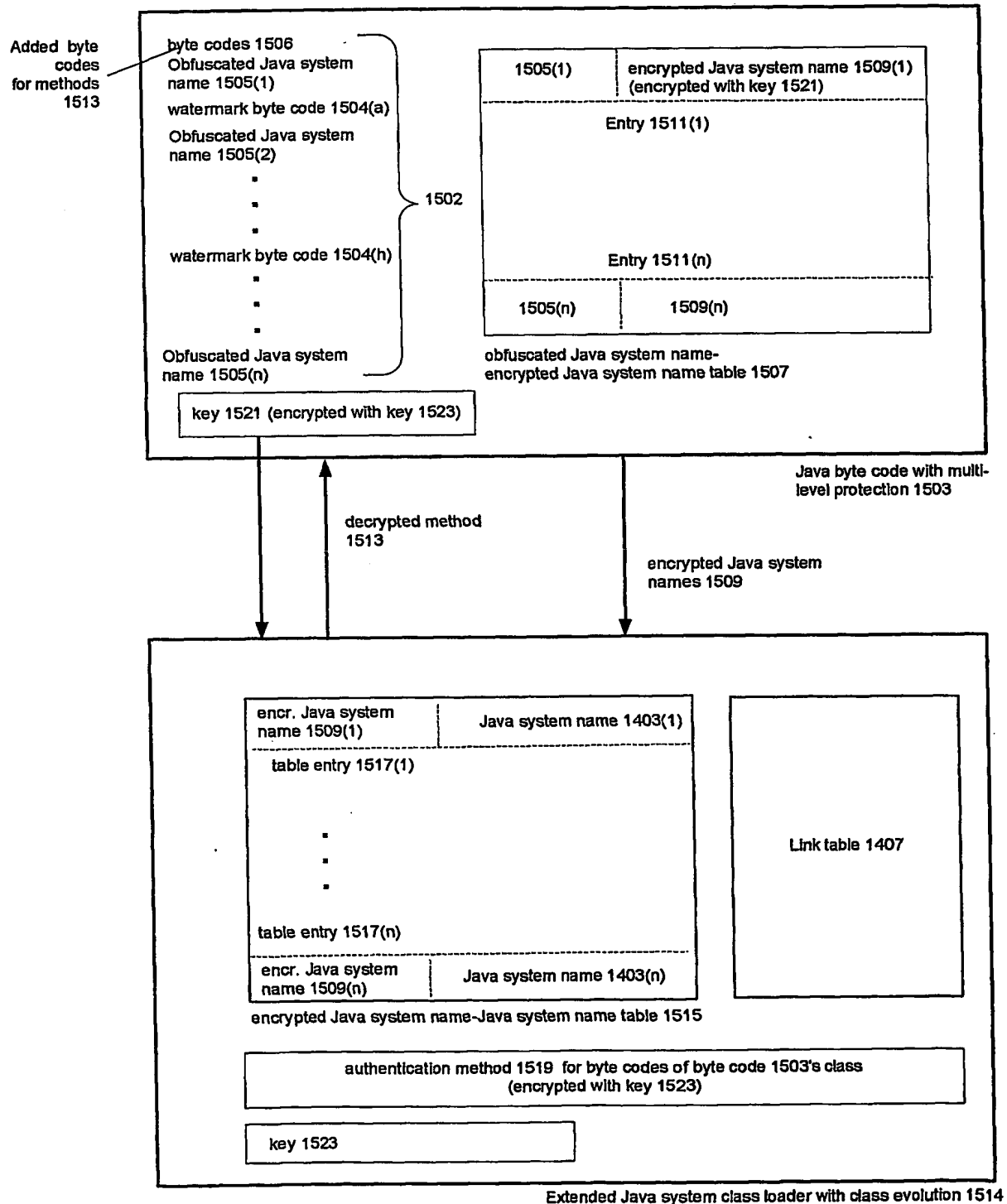


Fig. 14 Prior Art



1501

Fig. 15

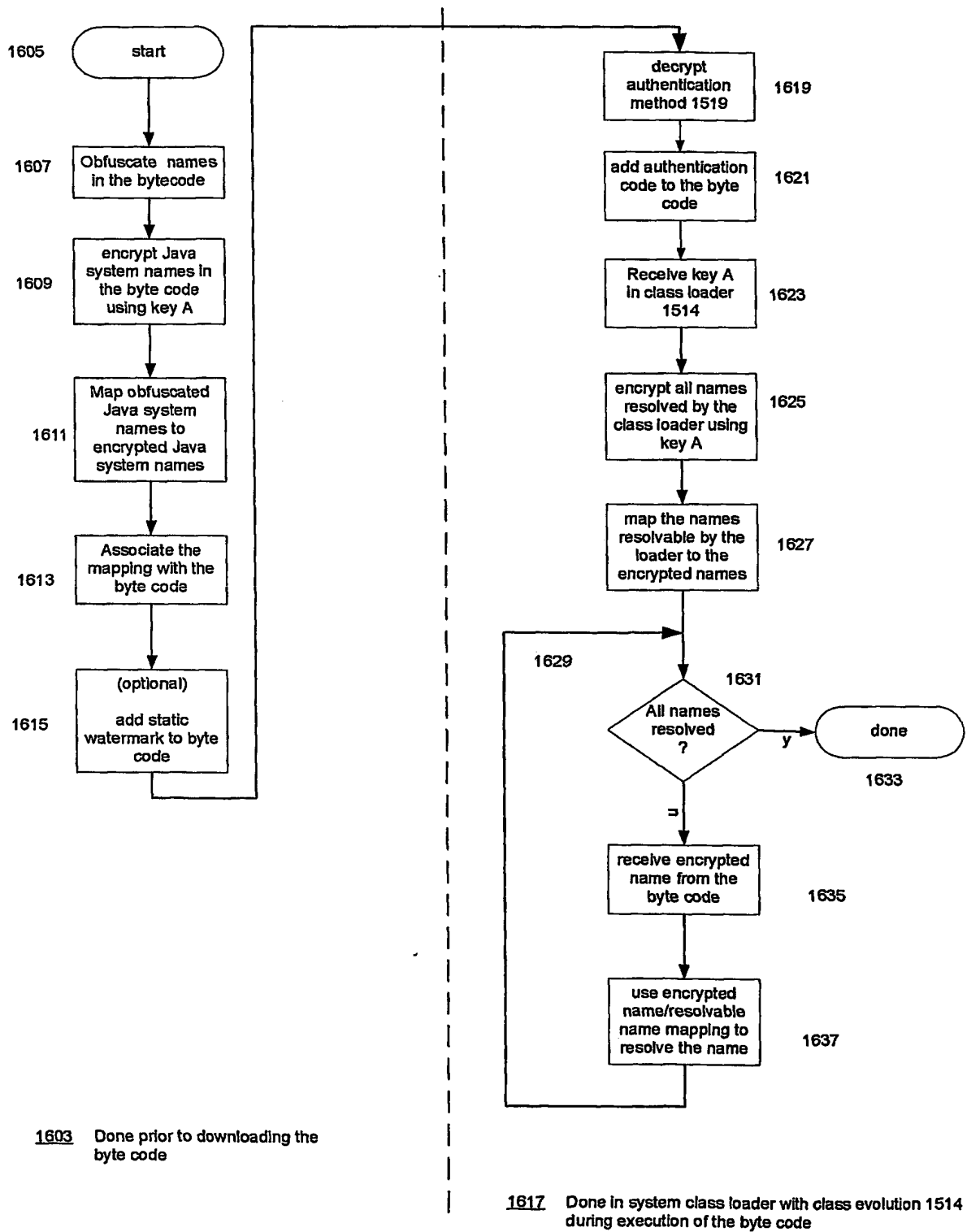


Fig. 16

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☒ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☒ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.